

**MAINTAINING A DOUBLE-ENDED QUEUE AS A LINKED-LIST WITH
SENTINEL NODES AND DELETE FLAGS WITH CONCURRENT NON-
BLOCKING INSERT AND REMOVE OPERATIONS USING A DOUBLE
COMPARE-AND-SWAP PRIMITIVE**

Nir N. Shavit,
Paul A. Martin, and
Guy L. Steele, Jr.

ABSTRACT OF THE DISCLOSURE

[1083] A linked-list-based concurrent shared object implementation has been developed that provides non-blocking and linearizable access to the concurrent shared object. In an application of the underlying techniques to a deque, the linked-list-based algorithm allows non-blocking completion of access operations without restricting concurrency in accessing the deque's two ends. The new implementation is based at least in part on a new technique for splitting a pop operation into two steps, marking that a node is about to be deleted, and then deleting it. Once marked, the node is logically deleted, and the actual deletion from the list can be deferred. In one realization, actual deletion is performed as part of a next push or pop operation performed at the corresponding end of the deque. An important aspect of the overall technique is synchronization of delete operations when processors detect that there are only marked nodes in the list and attempt to delete one or more of these nodes concurrently from both ends of the deque.